CA3 - Write Up

For this CA, myself and Ryan Jeffares collaborated to create a generative piece of music with strong ambient and IDM influences. From the outset, the main focus for this piece was to create an interesting piece of music with Pure Data, with the goal of implementing some sort of network protocol to allow us to sync up the patches in real time, while over halfway across the country from each other.

Although we tried using UDP, the protocol we decided on was TCP, due to its more stable nature and the better performance we experienced. Setting this up was a relatively easy task, involving each of us forwarding 2 ports in our routers, one for sending and one for recieving. To pack the data into a more organised format, we used OSC, specifically the packOSC and unpackOSC objects, which made it very easy to label the data which we were sending to each other.

We experimented with sending real-time audio to each other, however it was prety much impossible to get a useable signal with the low latency we needed, so this was abandoned. During the performance and our practice sessions, however, we used Discord (a VoIP service) in combination with Soundflower to stream audio so that we could hear each other. Although this still produces sub-satisfactory results, it was enough for us to be aware what was going on on the other user's computer.

The values that we sent each other were all generated from the sensors on our Arduinios. For me, this included a Pot controlling Ryan's FM frequency, a button that could trigger the FM synth to start generating notes, a pressure sensor that both modified the pitch of the drone on my end, but also controlled the speed of random impulses that triggered Ryan's pluck synth on his end. Finally, I also included a nice mechanical button to move the pitch of Ryan's pluck synth up an octave.

On my end, both sections of the piece have an instrument that could be described as a 'master instrument', in that they need tobe on in order to generate values for other instruments in that section, in order to synchronise time and pitch (although traditional harmony was pretty much completely ignored). In the first section, this instrument was the drone instrument, which controlled the pitch of the noise instrument. In the second section, this was the kick instrument, which randomly selects a time division, divides the overall tempo (in ms) by the time division selected, and passes the final value as the tempo for the percussion instrument, and the snare instrument.

The synthesis used in all of my instruments was pretty simple. The drone was made using FM synthesis, with a saw wave and a square wave. This was accompanied by a sine wave acting as a sub, unaffected by FM. These then go into a low-pass filter whose cutoff frequency is controlled by Ryan's arduino. The notes for this drone are randomly generated midi notes every 20000ms. These values then have portamento applied through the use of a lowpass filter. During the transition section however, the portamento and note generation are sped up, and the pitch is multiplied by 1.75.

The randomnotes which are triggered by a buttonon Ryan's arduino is even more simple. These are completely random pitch values, witha random envelope generatred using vline~. The synthesis for this is also FM, and uses 3 sine oscillators and a saw wave oscillator.

Finally for this section, the Noise instrument was made using 2 instances of a noise generator from the else library called 'gbman~'. The first instance is 32 times the frequency of the drone and lives in the left channel, and the second is 64 times the frequency of the drone and lives in the right channel. Both of these are put through a high-pass filter in order to reduce mudiness, as well as having their amplitude modulated by a brown noise generator to give a somewhat crunchy and glitchy effect.

The transition section is triggered by a button on Ryan's Arduino and (as mentioned previously) speeds up the portamento and note generation of the drone. It also triggers the percussion instrument to turn itself on and off a few times. This lasts until a counter has reached it's maximum value (28 beats from a 400ms metro). The drone is then automatically muted and a sample lifted from the Aphex Twin track 'Cock/ver10' plays. The sample delightfully requests a more interesting, fast moving section involving 303-inspired basslines in the words "Come on, you c**t, let's have some Aphex acid!!". This sample was only chosen because we thought it was funny. After the sample plays, a bang is sent from the player object and triggers to drums to begin playing, and Ryan's acid bass to begin playing on his end. The first beat of this section is the only place of reference in the piece that we can use to sync up the two videos.

The following section is, like most things in the piece, almost entirely randomly generated. The drum hits are triggered using a system based in the kick drum instrument. This randomly selects a time division (0.25, 0.33, 0.5, 0.75, or 1) and multiplies this by the overall tempo (400ms). This value is then passed into each instrument's metronome and used to set the speed of impulses. The actual sample chosen is also decided randomly. The range of numbers depends on the amount of samples chosen for each instrument. The random number is then passed into a messagebox containing "open samples/BD/$1.wav" in the case of the bass drum. This system was used for all three, however the percussion instrument also generates its own random pan values.

For graphics, although not discussed, one of us created visuals fitting for one of the two sections. My graphics were created for the second section, and consists of 3 parts,one for each drum instrument. The background is a small coloured square, whos colour changes to a random colour when the percussion instrument plays a sample. There is also a line-drawn cube which moves from the back of the screen to behind the camera every time a kick drum is triggered, and finally a line drawn sphere which is constantly rotating (using an LFO). This sphere appears in a different place in the 3D space each time a snare drum is triggered, along with changing the number of polygons used to make up the shape each time and it fades out into nothing very quickly.

I actually really hated PD before this assignment, due to how buggy it is and it’s frustrating lack of eror messages displaying any useful information at all, however working with PD to create something that I’m actually happy with, and finding network communication an easier task to accomplish than in Csound, I’ve warmed to the idea of working with PD, especially in the area of graphics.

Also, It’s named after Elon Musk’s new kid (X Æ A-12).